We claim:

1. A computer program product embodied on computer readable media readable by a computing system in a computing environment, for enforcing security policy using style sheet processing, comprising:

an input document;

one or more stored policy enforcement objects, wherein each of said stored policy enforcement objects specifies a security policy to be associated with zero or more elements of said input document;

a Document Type Definition (DTD) corresponding to said input document, wherein said DTD has been augmented with one or more references to selected ones of said stored policy enforcement objects;

an augmented style sheet processor, wherein said augmented processor further comprises:

computer-readable program code means for loading said DTD;

computer-readable program code means for resolving each of said one or more references in said loaded DTD;

computer-readable program code means for instantiating said policy enforcement objects associated with said resolved references;

computer-readable program code means for executing selected ones of said instantiated policy enforcement objects during application of one or more style sheets to said input document, wherein a result of said computer-readable program code means for executing is an interim transient document reflecting said execution;

computer-readable program code means for generating one or more random encryption keys;

computer-readable program code means for encrypting selected elements of said interim transient document, wherein a particular one of said generated random encryption keys may be used to encrypt one or more of said selected elements, while leaving zero or more other elements of said interim transient document unencrypted;

computer-readable program code means for encrypting each of said one or more random encryption keys; and

computer-readable program code means for creating an encrypted output document comprising said zero or more other unencrypted elements, said selected encrypted elements, and said encrypted encryption keys;

computer-readable program code means for receiving said encrypted output document at a client device;

an augmented document processor, comprising computer-readable program code means for decrypting said received output document for an individual user or process on said client device, thereby creating a result document; and

computer-readable program code means for rendering said result document on said client device.

2. The computer program product according to Claim 1, wherein said interim transient document comprises one or more encryption tags identifying elements needing encryption.

3. The computer program product according to Claim 1, wherein said input document is specified in an Extensible Markup Language (XML) notation.

4. The computer program product according to Claim 3, wherein said output document is specified in said XML notation.

5. The computer program product according to Claim 1, wherein said stored policy enforcement objects further comprise computer-readable program code means for overriding a method for evaluating said elements of said input document, and wherein said computer-readable program code means for executing further comprises computer-readable program code means for executing said computer-readable program code means for overriding.

6. The computer program product according to Claim 5, wherein said style sheets are specified in an Extensible Stylesheet Language (XSL) notation.

7. The computer program product according to Claim 6, wherein said method is a value-of method of said XSL notation, and wherein said computer-readable program code means for overriding said value-of method is by subclassing said value-of method.

8. The computer program product according to Claim 5 or Claim 7, wherein:

said overridden method comprises:

computer-readable program code means for generating encryption tags; and

computer-readable program code means for inserting said generated encryption tags into said interim transient document to surround elements of said interim transient document which are determined to require encryption; and

said computer-readable program code means for encrypting selected elements encrypts those elements surrounded by said inserted encryption tags.

9. The computer program product according to Claim 1, wherein each of said instantiated policy enforcement objects further comprises:

a specification of a community that is authorized to view said elements associated with said security policy; and

an encryption requirement for said elements associated with said security policy.

10. The computer program product according to Claim 9, wherein said encryption requirement further comprises specification of an encryption algorithm.

11. The computer program product according to Claim 9, wherein said encryption requirement further comprises specification of an encryption algorithm strength value.

12. The computer program product according to Claim 9, wherein:

said computer-readable program code means for encrypting said encryption keys further comprises computer-readable program code means for encrypting a different version of each of said random encryption keys for each of one or more members of each of zero or more of said

communities which uses said encryption key, and wherein each of said different versions is encrypted using a public key of said community member for which said different version was encrypted.

13. The computer program product according to Claim 9, wherein said encryption requirement may have a null value to indicate that said specified security policy does not require encryption.

14. The computer program product according to Claim 1, wherein said computer-readable program code means for encrypting selected elements uses a cipher block chaining mode encryption process.

15. The computer program product according to Claim 12, further comprising:

computer-readable program code means for creating a key class for each unique community, wherein said key class is associated with each of said encrypted elements for which this unique community is an authorized viewer, and wherein said key class comprises: (1) a strongest encryption requirement of said associated encrypted elements; (2) an identifier of each member of said unique community; and (3) one of said different versions of said encrypted encryption key for each of said identified community members; and

wherein:

said computer-readable program code means for generating said one or more random encryption keys generates a particular one of said random encryption keys for each of

said key classes, and wherein each of said different versions in a particular key class is encrypted from said generated encryption key generated for said key class; and

said computer-readable program code means for encrypting selected elements uses that one of said particular random encryption keys which was generated for said key class with which said selected element is associated.

16. The computer program product according to Claim 12, wherein:

said computer-readable program code means for decrypting said output document further comprises:

computer-readable program code means for determining zero or more of said communities of which said individual user or process is one of said members;

computer-readable program code means for decrypting, for each of said determined communities, said different version of said random encrytion key which was encrypted using said public key of said one member, wherein said computer-readable program code means for decrypting uses a private key of said one member which is associated with said public key which was used for encryption, thereby creating a decrypted key; and

computer-readable program code means for decrypting selected ones of said encrypted elements in said output document using said decrypted keys, wherein said selected ones of said encrypted elements are those which were encrypted for one of said determined communities; and

said computer-readable program code means for rendering further comprises:

computer-readable program code means for rendering said decrypted selected ones and said other unencrypted elements.

17. The computer program product according to Claim 15, wherein:

said computer-readable program code means for decrypting said output document further comprises:

computer-readable program code means for determining zero or more of said key classes which identify said individual user or process as one of said members;

computer-readable program code means for decrypting, for each of said determined key classes, said different version of said random encrytion key in said key class which was encrypted using said public key of said one member, wherein said computer-readable program code means for decrypting uses a private key of said one member which is associated with said public key which was used for encryption, thereby creating a decrypted key; and

computer-readable program code means for decrypting selected ones of said encrypted elements in said output document using said decrypted keys, wherein said selected ones of said encrypted elements are those which were encrypted for said key class; and

said computer-readable program code means for rendering further comprises:

computer-readable program code means for rendering said decrypted selected ones and said other unencrypted elements.

18. The computer program product according to Claim 16 or Claim 17, wherein said computer-readable program code means for rendering further comprises computer-readable

program code means for rendering a substitute text message for any of said selected encrypted elements in said output document which cannot be decrypted by said computer-readable program code means for decrypting said output document.

19. The computer program product according to Claim 1, wherein said DTD is replaced by a schema.

20. The computer program product according to Claim 9, wherein said encryption requirement further comprises specification of an encryption key length.

21. The computer program product according to Claim 8, wherein said inserted encryption tags may surround either values of said elements or values and tags of said elements.

22. A system for enforcing security policy using style sheet processing in a computing environment, comprising:

an input document;

one or more stored policy enforcement objects, wherein each of said stored policy enforcement objects specifies a security policy to be associated with zero or more elements of said input document;

a Document Type Definition (DTD) corresponding to said input document, wherein said DTD has been augmented with one or more references to selected ones of said stored policy enforcement objects;

an augmented style sheet processor, wherein said augmented processor further comprises:

means for loading said DTD;

means for resolving each of said one or more references in said loaded DTD;

means for instantiating said policy enforcement objects associated with said resolved references;

means for executing selected ones of said instantiated policy enforcement objects during application of one or more style sheets to said input document, wherein a result of said means for executing is an interim transient document reflecting said execution;

means for generating one or more random encryption keys;

means for encrypting selected elements of said interim transient document, wherein a particular one of said generated random encryption keys may be used to encrypt one or more of said selected elements, while leaving zero or more other elements of said interim transient document unencrypted;

means for encrypting each of said one or more random encryption keys; and

means for creating an encrypted output document comprising said zero or more other unencrypted elements, said selected encrypted elements, and said encrypted encryption keys;

means for receiving said encrypted output document at a client device;

an augmented document processor, comprising means for decrypting said received output document for an individual user or process on said client device, thereby creating a result document; and

means for rendering said result document on said client device.

23. The system according to Claim 22, wherein said interim transient document comprises one or more encryption tags identifying elements needing encryption.

24. The system according to Claim 22, wherein said input document is specified in an Extensible Markup Language (XML) notation.

25. The system according to Claim 24, wherein said output document is specified in said XML notation.

26. The system according to Claim 22, wherein said stored policy enforcement objects further comprise means for overriding a method for evaluating said elements of said input document, and wherein said means for executing further comprises means for executing said means for overriding.

27. The system according to Claim 26, wherein said style sheets are specified in an Extensible Stylesheet Language (XSL) notation.

28. The system according to Claim 27, wherein said method is a value-of method of said XSL notation, and wherein said means for overriding said value-of method is by subclassing said value-of method.

29. The system according to Claim 26 or Claim 28, wherein:

said overridden method comprises:

means for generating encryption tags; and

means for inserting said generated encryption tags into said interim transient document to surround elements of said interim transient document which are determined to require encryption; and

said means for encrypting selected elements encrypts those elements surrounded by said inserted encryption tags.

30. The system according to Claim 22, wherein each of said instantiated policy enforcement objects further comprises:

a specification of a community that is authorized to view said elements associated with said security policy; and

an encryption requirement for said elements associated with said security policy.

31. The system according to Claim 30, wherein said encryption requirement further comprises specification of an encryption algorithm.

32. The system according to Claim 30, wherein said encryption requirement further comprises specification of an encryption algorithm strength value.

33. The system according to Claim 30, wherein:

said means for encrypting said encryption keys further comprises means for encrypting a different version of each of said random encryption keys for each of one or more members of each of zero or more of said communities which uses said encryption key, and wherein each of said different versions is encrypted using a public key of said community member for which said different version was encrypted.

34. The system according to Claim 30, wherein said encryption requirement may have a null value to indicate that said specified security policy does not require encryption.

35. The system according to Claim 22, wherein said means for encrypting selected elements uses a cipher block chaining mode encryption process.

36. The system according to Claim 33, further comprising:

means for creating a key class for each unique community, wherein said key class is associated with each of said encrypted elements for which this unique community is an authorized viewer, and wherein said key class comprises: (1) a strongest encryption requirement of said associated encrypted elements; (2) an identifier of each member of said unique community; and (3) one of said different versions of said encrypted encryption key for each of said identified community members; and

wherein:

said means for generating said one or more random encryption keys generates a particular one of said random encryption keys for each of said key classes, and wherein each of

said different versions in a particular key class is encrypted from said generated encryption key generated for said key class; and

said means for encrypting selected elements uses that one of said particular random encryption keys which was generated for said key class with which said selected element is associated.

37. The system according to Claim 33, wherein:

said means for decrypting said output document further comprises:

means for determining zero or more of said communities of which said individual user or process is one of said members;

means for decrypting, for each of said determined communities, said different version of said random encrytion key which was encrypted using said public key of said one member, wherein said means for decrypting uses a private key of said one member which is associated with said public key which was used for encryption, thereby creating a decrypted key; and

means for decrypting selected ones of said encrypted elements in said output document using said decrypted keys, wherein said selected ones of said encrypted elements are those which were encrypted for one of said determined communities; and

said means for rendering further comprises:

means for rendering said decrypted selected ones and said other unencrypted elements.

38. The system according to Claim 36, wherein:

said means for decrypting said output document further comprises:

means for determining zero or more of said key classes which identify said individual user or process as one of said members;

means for decrypting, for each of said determined key classes, said different version of said random encrytion key in said key class which was encrypted using said public key of said one member, wherein said means for decrypting uses a private key of said one member which is associated with said public key which was used for encryption, thereby creating a decrypted key; and

means for decrypting selected ones of said encrypted elements in said output document using said decrypted keys, wherein said selected ones of said encrypted elements are those which were encrypted for said key class; and

said means for rendering further comprises:

means for rendering said decrypted selected ones and said other unencrypted elements.

39. The system according to Claim 37 or Claim 38, wherein said means for rendering further comprises means for rendering a substitute text message for any of said selected encrypted elements in said output document which cannot be decrypted by said means for decrypting said output document.

40. The system according to Claim 22, wherein said DTD is replaced by a schema.

41. The system according to Claim 30, wherein said encryption requirement further comprises specification of an encryption key length.

42. The system according to Claim 29, wherein said inserted encryption tags may surround either values of said elements or values and tags of said elements.

43. A method for enforcing security policy using style sheet processing in a computing environment, comprising the steps of:

providing an input document;

providing one or more stored policy enforcement objects, wherein each of said stored policy enforcement objects specifies a security policy to be associated with zero or more elements of said input document;

providing a Document Type Definition (DTD) corresponding to said input document, wherein said DTD has been augmented with one or more references to selected ones of said stored policy enforcement objects;

executing an augmented style sheet processor, further comprising the steps of:

loading said DTD;

resolving each of said one or more references in said loaded DTD;

instantiating said policy enforcement objects associated with said resolved references;

executing selected ones of said instantiated policy enforcement objects during application of one or more style sheets to said input document, wherein a result of said executing selected ones step is an interim transient document reflecting said execution;

generating one or more random encryption keys;

encrypting selected elements of said interim transient document, wherein a particular one of said generated random encryption keys may be used to encrypt one or more of said selected elements, while leaving zero or more other elements of said interim transient document unencrypted;

encrypting each of said one or more random encryption keys; and

creating an encrypted output document comprising said zero or more other unencrypted elements, said selected encrypted elements, and said encrypted encryption keys;

receiving said encrypted output document at a client device;

executing an augmented document processor, comprising the step of decrypting said received output document for an individual user or process on said client device, thereby creating a result document; and

rendering said result document on said client device.

44. The method according to Claim 43, wherein said interim transient document comprises one or more encryption tags identifying elements needing encryption.

45. The method according to Claim 43, wherein said input document is specified in an Extensible Markup Language (XML) notation.

46. The method according to Claim 45, wherein said output document is specified in said XML notation.

47. The method according to Claim 43, wherein said stored policy enforcement objects further comprise executable code for overriding a method for evaluating said elements of said input document, and wherein said executing selected ones step further comprises overriding said method for evaluating.

48. The method according to Claim 47, wherein said style sheets are specified in an Extensible Stylesheet Language (XSL) notation.

49. The method according to Claim 48, wherein said method is a value-of method of said XSL notation, and wherein said step of overriding said value-of method is by subclassing said value-of method.

50. The method according to Claim 47 or Claim 49, wherein:

said step of overriding further comprises the steps of:

generating encryption tags; and

inserting said generated encryption tags into said interim transient document to surround elements of said interim transient document which are determined to require encryption; and

said step of encrypting selected elements encrypts those elements surrounded by said inserted encryption tags.

51. The method according to Claim 43, wherein each of said instantiated policy enforcement objects further comprises:

a specification of a community that is authorized to view said elements associated with said security policy; and

an encryption requirement for said elements associated with said security policy.

52. The method according to Claim 51, wherein said encryption requirement further comprises specification of an encryption algorithm.

53. The method according to Claim 51, wherein said encryption requirement further comprises specification of an encryption algorithm strength value.

54. The method according to Claim 51, wherein:

said step of encrypting said encryption keys further comprises the step of encrypting a different version of each of said random encryption keys for each of one or more members of each of zero or more of said communities which uses said encryption key, and wherein each of said different versions is encrypted using a public key of said community member for which said different version was encrypted.

55. The method according to Claim 51, wherein said encryption requirement may have a null value to indicate that said specified security policy does not require encryption.

56. The method according to Claim 43, wherein said step of encrypting selected elements uses a cipher block chaining mode encryption process.

57. The method according to Claim 54, further comprising the step of:

creating a key class for each unique community, wherein said key class is associated with each of said encrypted elements for which this unique community is an authorized viewer, and wherein said key class comprises: (1) a strongest encryption requirement of said associated encrypted elements; (2) an identifier of each member of said unique community; and (3) one of said different versions of said encrypted encryption key for each of said identified community members; and

wherein:

said step of generating said one or more random encryption keys generates a particular one of said random encryption keys for each of said key classes, and wherein each of said different versions in a particular key class is encrypted from said generated encryption key generated for said key class; and

said step of encrypting selected elements uses that one of said particular random encryption keys which was generated for said key class with which said selected element is associated.

58. The method according to Claim 54, wherein:

said step of decrypting said output document further comprises the steps of:

determining zero or more of said communities of which said individual user or process is one of said members;

decrypting, for each of said determined communities, said different version of said random encrytion key which was encrypted using said public key of said one member, wherein said step of decrypting uses a private key of said one member which is associated with said public key which was used for encryption, thereby creating a decrypted key; and

decrypting selected ones of said encrypted elements in said output document using said decrypted keys, wherein said selected ones of said encrypted elements are those which were encrypted for one of said determined communities; and

said step of rendering further comprises the step of:

rendering said decrypted selected ones and said other unencrypted elements.

59. The method according to Claim 57, wherein:

said step of decrypting said output document further comprises the steps of:

determining zero or more of said key classes which identify said individual user or process as one of said members;

decrypting, for each of said determined key classes, said different version of said random encrytion key in said key class which was encrypted using said public key of said one member, wherein said step of decrypting uses a private key of said one member which is

associated with said public key which was used for encryption, thereby creating a decrypted key; and

decrypting selected ones of said encrypted elements in said output document using said decrypted keys, wherein said selected ones of said encrypted elements are those which were encrypted for said key class; and

said step of rendering further comprises the step of:

rendering said decrypted selected ones and said other unencrypted elements.

60. The method according to Claim 58 or Claim 59, wherein said step of rendering further comprises the step of rendering a substitute text message for any of said selected encrypted elements in said output document which cannot be decrypted by said step of decrypting said output document.

61. The method according to Claim 43, wherein said DTD is replaced by a schema.

62. The method according to Claim 51, wherein said encryption requirement further comprises specification of an encryption key length.

63. The method according to Claim 50, wherein said inserted encryption tags may surround either values of said elements or values and tags of said elements.